



Ordenación #2

Transcripción

[00:00] Hay una última cosa que les quiero enseñar con respecto a esta clase. Por ejemplo, ya estamos viendo que podemos decirle a Spring que queremos que nos retorne la base de datos dependiendo de lo que lo enviemos.

[00:17] Pero si por ejemplo alguna vez se topan con un problema y que necesitan ver las queries que Spring está haciendo, o bueno, mejor dicho, que JPA está ejecutado sobre la base de datos, yo puedo actualmente mostrar lo que está sucediendo.

[00:34] Aquí en los logs ahora yo no tengo ninguna información de lo que pasa con la base de datos. ¿Qué hago para mostrar esa información? Bueno, Spring, como siempre, nos da la facilidad de configurar muchas cosas y aquí voy a darle un `spring.jpa` y el parámetro es `show-sql` que vemos que por defecto está `post` y le voy a dar `true`.

[01:02] Vamos a guardar esta configuración, esperamos que el servidor reinicie. Con esto. ¿qué les quiero decir? Simplemente, Spring muéstrame los parámetros SQL o las queries SQL que se están ejecutando entre la aplicación y la base de datos. Como esto fue una modificación a nivel de las propiedades, yo voy a reiniciar el servidor manualmente. Vamos a esperar a que cargue.

[01:31] Vemos que cargó y vamos a ejecutar nuevamente esto aquí. Vamos aquí y vemos en efecto que dice Hibernate. ¿Qué hizo Hibernate? Hizo un `select`. Hizo el `select from médicos, limit`, algún parámetro, etcétera. Esos son los

parámetros que lleguen, este es el prepared statement que ya Hibernate hace por nosotros.

[01:54] Y hace una query más, entonces nos dimos cuenta que ir Hibernate aquí se ejecuta dos queries diferentes en la cual ejecuta un count, donde médicos igual, todo esto que vemos aquí son alias, son alias que Hibernate le da a cada parámetro.

[02:13] Internamente él ya mapea estos alias y nos crea la query dinámicamente. Pero de todas formas, personalmente a mí me parece muy difícil de leer. Imagínense si tuviéramos 15 campos en la base de datos, en nuestra entidad. No es un caso de uso difícil de encontrar en el mercado. 15 campos en una tabla es algo normal.

[02:38] Imagínense hasta dónde llegaría la query. Sería muy difícil de leer. Entonces, existe también otra forma de formatear las queries que tenemos aquí y sería esta de aquí. `Spring.jpa.hibernate`, sería `spring.jpa`, no `database`, sería `.properties.hibernate` y aquí vemos que dentro de Hibernate yo tengo algunas otras propiedades.

[03:14] Y la que yo voy a usar es `format_sql`. Y le voy a decir que sí quiero que formatees el SQL que estás generando y eso es propio de Hibernate, porque ya vimos que quien genera las queries es Hibernate como implementación de JPA.

[03:33] JPA le da la especificación, Hibernate es la implementación, es quien hace el trabajo serio. Iniciamos nuevamente nuestro servidor. Vemos que inicia correctamente, limpiamos esto. Y ejecutamos nuevamente la query. Nada. No hay diferencias aquí, pero la diferencia ya viene aquí.

[03:57] Observen ahora cómo está formateada la query con Hibernate. Vemos que es exactamente la misma que vimos hace un rato, pero es mucho más fácil de leer, y si estás haciendo algún debug o algún análisis de algún issue, sin duda esto sí te puede ayudar mucho.

[04:18] No es recomendable tener todo esto en sistemas de producción, porque bueno, te va a costar un poco de performance y tus logs se van a ver muy contaminados con un montón de queries, pero si estás haciendo desarrollo a nivel de tu máquina y quieres saber qué es lo que está pasando.

[04:35] Si quieres resolver un tema de performance, yo estoy seguro que estos dos parámetros de aquí, estas dos configuraciones te van a ayudar mucho para entender lo que Spring hace internamente. En este caso lo que Hibernate hace internamente.

[04:49] La próxima clase va a estar mucho más interesante, vamos a aprender a editar y eliminar registros, entonces nos vemos.